In the claims:

1. (currently amended) Method for preserving security associations between at least two entities ~~include~~ comprising the steps of:

maintaining a security association relating to communication between the at least two entities in ~~a table~~ in volatile storage;

~~, and periodically~~ storing a copy of the security association in non-volatile storage; and

in response to detection of corruption of the security association in volatile storage, where the corruption is caused by an event other than power failure, employing the copy of the security association in non-volatile storage to update the security association in volatile storage.

2. (currently amended) The method according to claim 1, further comprising the step of encrypting the security association prior to ~~periodically~~ storing the security association in the nonvolatile storage.

3. (currently amended) The method according to claim 1 wherein the step of ~~periodically~~ storing includes the step of detecting a trigger event.

4. (previously presented) The method according to claim 3 wherein the step of detecting a trigger event includes the step of detecting a change in the security association.

5. (currently amended) The method according to claim 1 further comprising the step of updating the contents of ~~the~~ a security associations table using the security association stored in non-volatile storage.

6. (currently amended) A method for maintaining security associations between a server and a member, the method comprising the steps of:

generating a security association permitting communication between the server and the member;

storing the security association in a location of volatile memory available to the server;

~~periodically~~ storing a copy of the security association in a non-volatile memory; and

retrieving the copy of the security association from the non-volatile memory in the event that the security association becomes unavailable to the server because of corruption of the security association in volatile memory caused by an event other than power failure, and employing the copy of the security association in non-volatile memory to update the security association in volatile memory.

7. (currently amended) The method of claim 6, further comprising the steps of encrypting the security association prior to the step of ~~periodically~~ storing the security association in the non-volatile memory.

8. (currently amended) The method of claim 6, wherein the step of ~~periodically~~ storing the security association includes the step of detecting a trigger event.

9. (previously presented) The method of claim 8, wherein the step of detecting the trigger event includes the step of detecting a new security association between the server and the member.

10. (currently amended) An apparatus for preserving security associations between at least two entities comprises:

a volatile memory including a first table for storing a security association related to communication between the at least two entities;

a non-volatile memory including a second table for storing at least a portion of the first table; ~~and~~

means for ~~periodically~~ copying the at least a portion of the first table to the second table; and

means for copying at least a portion of the second table to the first table in response to detection of corruption of the first table, where the corruption is caused by an event other than power failure.

11. (previously presented) The apparatus of claim 10, further comprising means for encrypting the at least a portion of the first table prior to copying the at least a portion of the first table to the second table.

12. (previously presented) The apparatus of claim 10 further comprising means for copying overwriting the at least a portion of the first table with contents of the second table.

13. (previously presented) The apparatus of claim 10 including encryption logic for encrypting the at least a portion of the first table.

14. (previously presented) The apparatus of claim 10 including decryption logic for decrypting the second table.

15. (previously presented) The apparatus of claim 10 further comprising a key, stored in non-volatile memory, for encrypting the at least a portion of the first table.